

Initiation à la programmation des systèmes embarqués

Découverte de la programmation des cartes Arduino

Courte Présentation

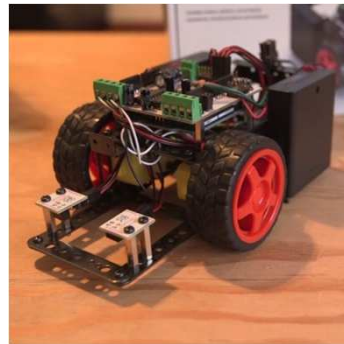
Une carte Arduino, pour quoi faire ?

- ... pour réaliser des (mini ?) projets



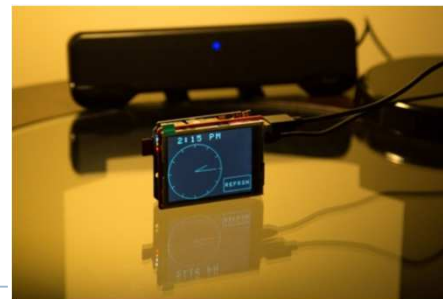
Harpe laser

<http://makezine.com/projects/laser-harp/>



Joute robotique

<http://makezine.com/video/ready-set-joust/>

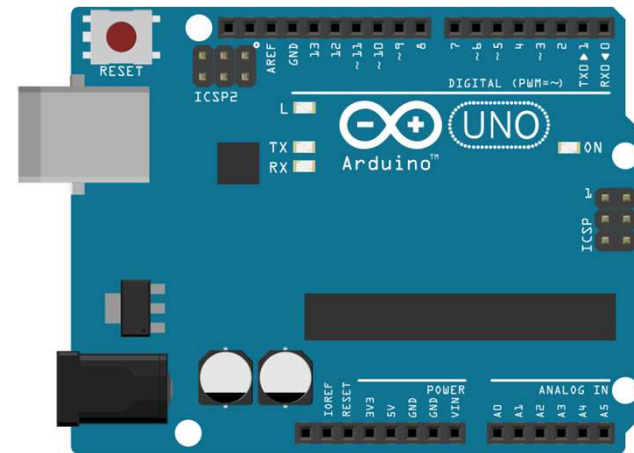
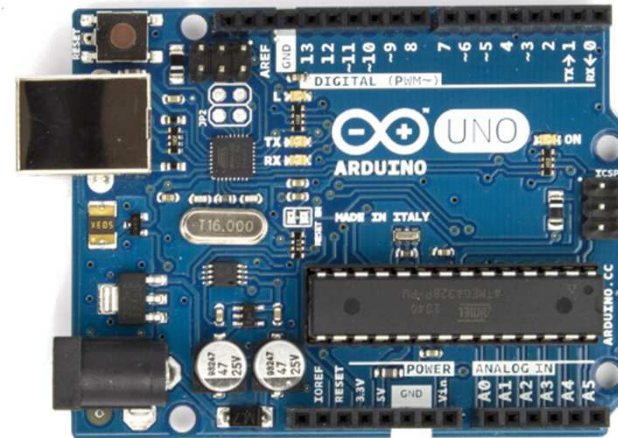


Réveil

<http://makezine.com/video/never-forget-to-set-an-alarm-because-this-alarm-clock-sets-itself/>

La carte Arduino Uno

- ▶ Microcontroller : ATmega328
- ▶ Operating Voltage : 5V
- ▶ Input Voltage (recommended) : 7-12V
- ▶ Input Voltage (limits) : 6-20V
- ▶ Digital I/O Pins : 14 (of which 6 provide PWM output)
- ▶ Analog Input Pins : 6
- ▶ DC Current per I/O Pin : 40 mA
- ▶ DC Current for 3.3V Pin : 50 mA
- ▶ Flash Memory : 32 KB (ATmega328) of which 0.5 KB used by bootloader
- ▶ SRAM : 2 KB (ATmega328)
- ▶ EEPROM : 1 KB (ATmega328)
- ▶ Clock Speed : 16 MHz
- ▶ Length : 68.6 mm
- ▶ Width : 53.4 mm
- ▶ Weight : 25 g

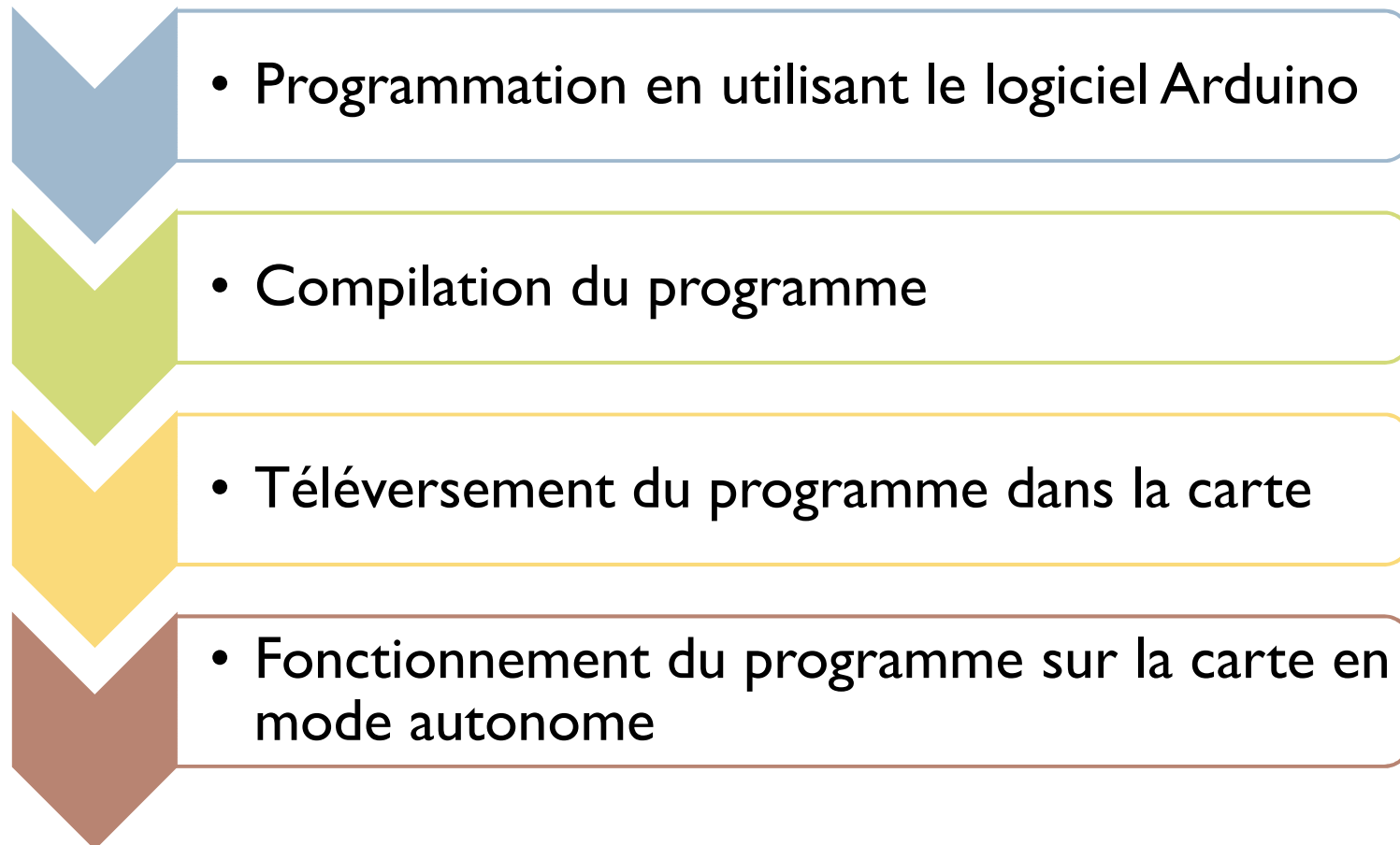


fritzing

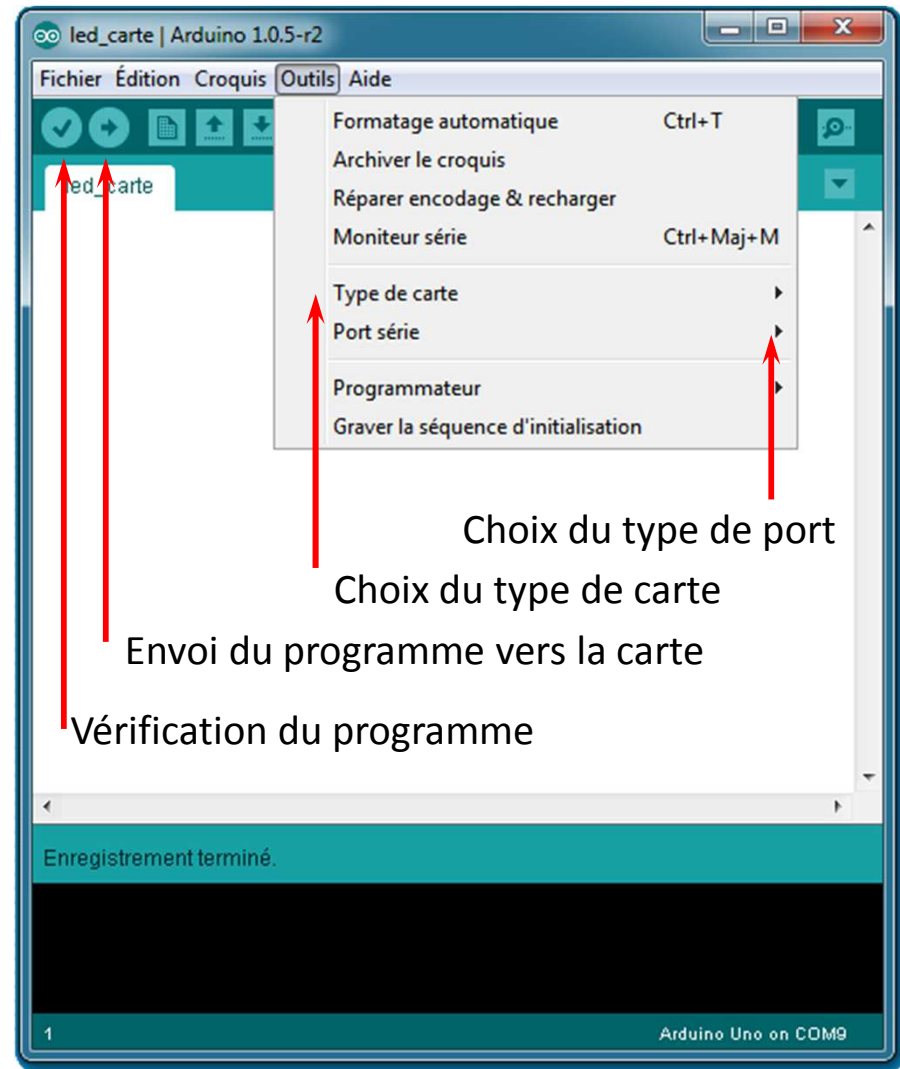
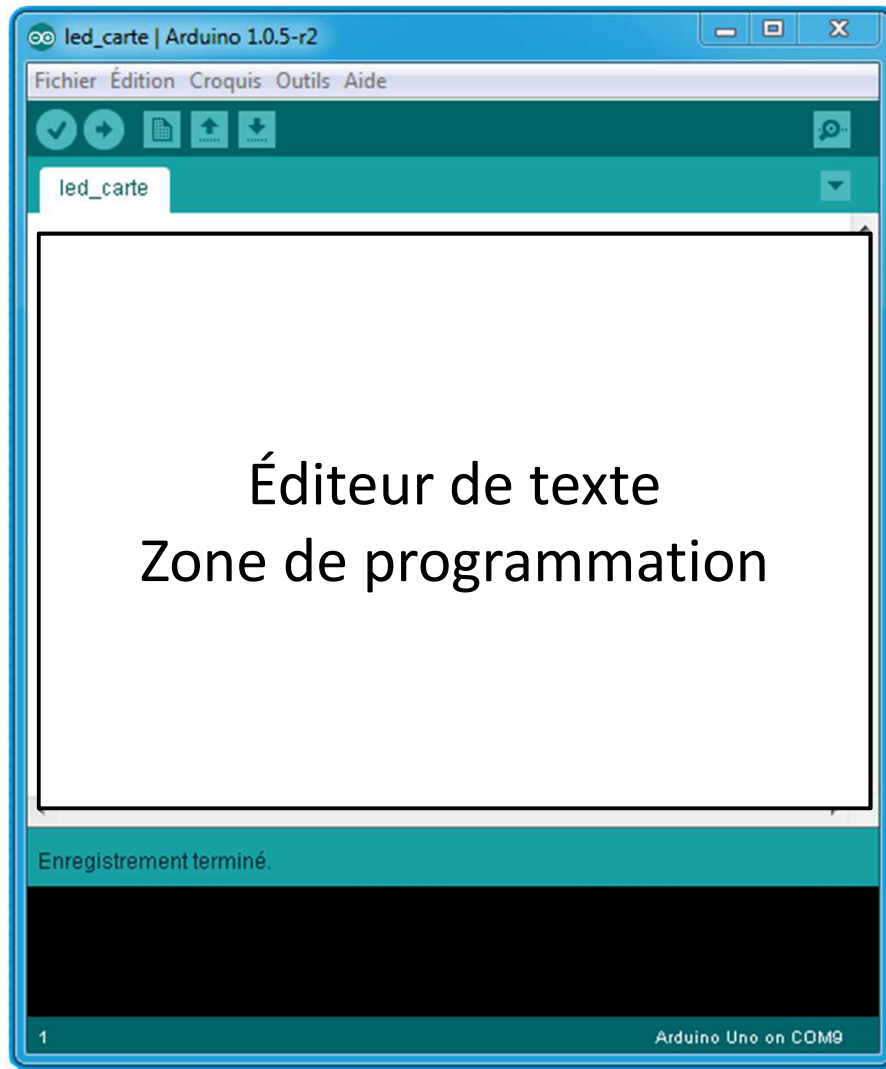
Programmation par le logiciel Arduino

THE programme de base : Faire clignoter la LED

Processus de réalisation du programme

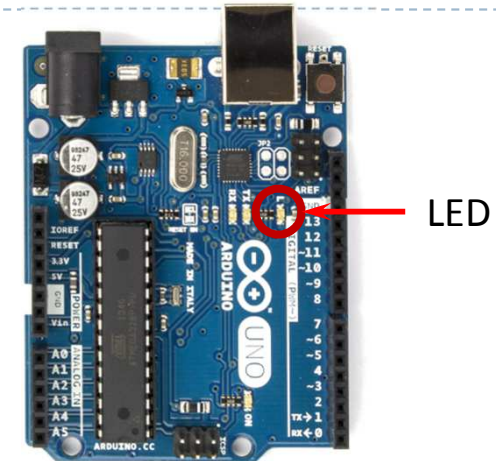


Interface du logiciel Arduino



Allumer et éteindre une Led par période de 1 seconde

- ▶ A la différence de Python, il faut déclarer les variables et leur type.
- ▶ Les lignes doivent se terminer par des
- ▶ Void désigne la déclaration d'une fonction
- ▶ Pour réaliser un commentaire il faut faire précéder le commentaire de //



```
int led = 13;
```

Déclaration de la variable entière led et affectation du nombre 13 à la variable led.
13 correspond à une sortie de la carte Arduino possédant une led.

```
void setup() {  
    pinMode(led, OUTPUT);  
}
```

Setup : configuration du matériel. Lancée après avoir appuyé sur Reset.
La broche led est déclarée comme une sortie

```
void loop() {  
    digitalWrite(led, HIGH);  
    delay(1000);  
    digitalWrite(led, LOW);  
    delay(1000);  
}
```

La boucle loop est lancée un nombre infini de fois.

Cette séquence permet d'allumer et d'éteindre la led toutes les secondes.

Travail à réaliser

► Activité 1 : clignotement de la LED

1. Situer la carte sur la led.
2. Saisir le code de la page précédente sur le logiciel Arduino .
3. Lancer la vérification .
4. Implanter le programme sur la carte.
5. Vérifier son bon fonctionnement.
6. Modifier le programme pour modifier le temps d'allumage et d'extinction.

► Activité 2 : clignotement de la LED – Affichage sur la console série (La console série permet d'afficher des informations à destination de l'utilisateur.

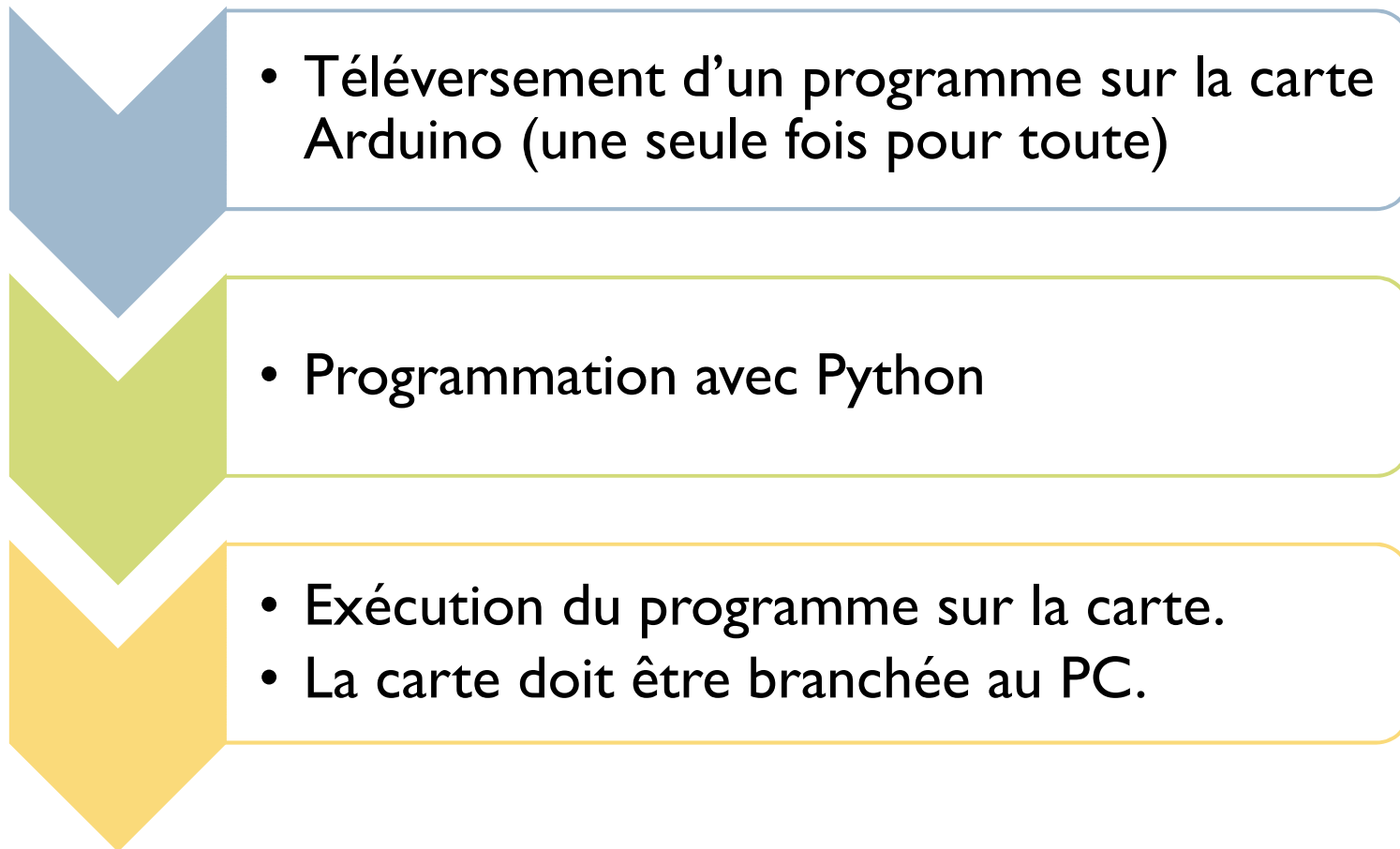
1. Modifier le code de l'activité précédente en utilisant le code ci-contre.
2. Pour afficher les messages :
 1. Menu outil
 2. Console série

```
int led = 13;
void setup() {
    pinMode(led, OUTPUT);
    Serial.begin(57600);
}
void loop() {
    digitalWrite(led, HIGH);
    Serial.print("JOUR \n");
    delay(1000);
    digitalWrite(led, LOW);
    Serial.print("NUIT \n");
    delay(1000);
}
```

Programmation par Python

THE programme de base (again) : Faire clignoter la LED

Processus de réalisation du programme



Préambule – Installation

- ▶ Dans le logiciel Arduino
 - ▶ Menu croquis ▶ Inclure une bibliothèque ▶ Ajouter la bibliothèque Zip
 - ▶ Ajouter la bibliothèque TimerOne-rll.zip pour l'installer
 - ▶ Sélectionner la carte Arduino Uno et le port COM sur lequel la carte est connectée.
 - ▶ Téléverser le programme toolbox_arduino_v4.ino
- ▶ Placer le programme py2duino.py dans le répertoire Lib
 - ▶ Exemple de notation du port : "/dev/tty3"

Travail à réaliser

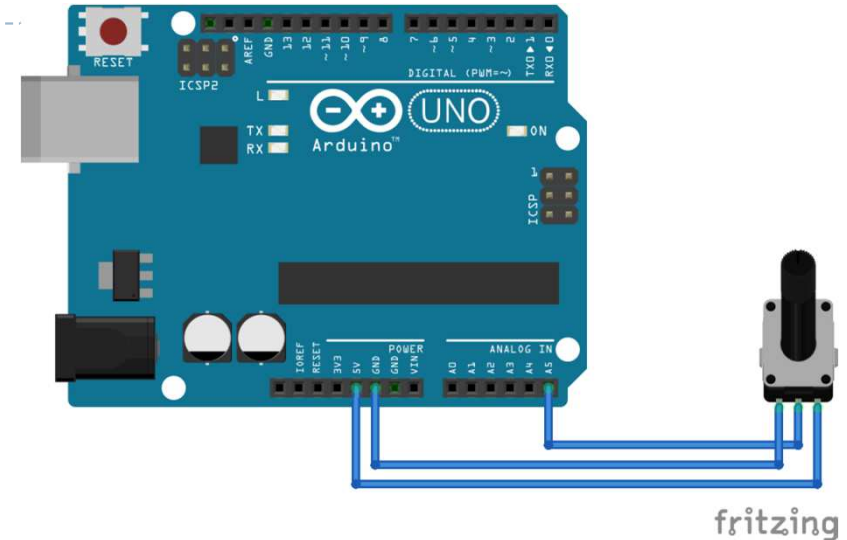
- ▶ **Téléversement préalable :**
 - ▶ Il est nécessaire de téléverser une seule fois le programme `toolbox_arduino_v4.ino` sur la carte. Ce programme permet à Python de communiquer avec la carte Arduino.
- ▶ **Activité 3 :**
 - ▶ Ouvrir le fichier `led.py` et analyser sa structure.
 - ▶ Exécuter le programme.
 - ▶ Observer la LED clignoter.
 - ▶ Modifier le programme pour que l'état de la LED soit affiché dans la console.

Programmation par Python

Acquisition du signal d'un potentiomètre rotatif

Activité 4 : Acquisition d'un signal analogique provenant d'un potentiomètre

- ▶ Matériel nécessaire
 - ▶ Une carte arduino
 - ▶ Un potentiomètre
 - ▶ Une planche à pain ;) (Bread bord)
- ▶ Réaliser le câblage ci-contre



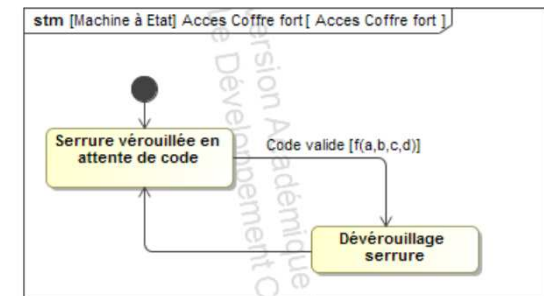
- ▶ Réaliser le programme Python permettant d'afficher sur la console l'information provenant du potentiomètre.
 - ▶ Déclaration d'une entrée analogique :
 - ▶ `AI = AnalogInput(MaCarte,PIN_ANALOGIQUE)`
 - ▶ Lecture de l'information analogique :
 - ▶ `AI.read()`

Programmation par Python

Un peu de logique combinatoire

Retour sur le coffre fort de banque

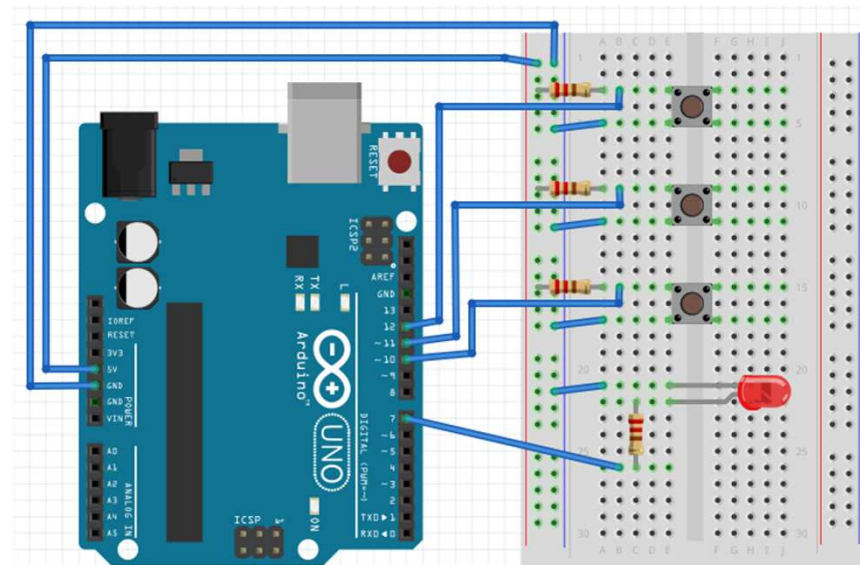
- ▶ Rappel :
- ▶ On s'intéresse à un coffre-fort de banque dont on donne le principe de fonctionnement.
- ▶ Seuls 4 responsables (notés A, B, C et D) qui possèdent un ensemble code d'accès + clef à serrure peuvent avoir accès au coffre. Le responsable A possède l'ensemble code d'accès et une clef notée a . Le responsable B possède l'ensemble code d'accès et une clef notée b . Le responsable C possède l'ensemble code d'accès et une clef notée c . Le responsable D possède l'ensemble code d'accès et une clef notée d .
- ▶ Le responsable A ne peut ouvrir le coffre qu'avec le responsable B ou C .
- ▶ Les responsables B, C et D ne peuvent ouvrir le coffre qu'en présence d'au moins deux des autres responsables.
- ▶ On montre que le coffre s'ouvre si on a :
- ▶ $S = b \cdot c \cdot d + a \cdot c + a \cdot b$



D'après TD de Florestan Mathurin

Implémentation du fonctionnement

- ▶ Chacune des 4 entrées sera représentée par un bouton poussoir. L'ouverture du coffre sera représenté par une LED.
- ▶ Matériel nécessaire :
 - ▶ 4 boutons poussoirs;
 - ▶ 4 résistances de 4,7 k Ω (Jaune, Violet, Rouge doré);
 - ▶ 1 led (verte) en série avec une résistance de 1 k Ω .
- ▶ Le câblage est le suivant



Activité finale

- ▶ Câbler un seul bouton poussoir
 - ▶ Réaliser le code Python permettant de connaître l'état du bouton. Que constatez-vous ? Commenter.
- ▶ Câbler la led.
 - ▶ Réaliser le code Python permettant d'allumer la LED lorsque le bouton est pressé.
- ▶ Réaliser le câblage final.
 - ▶ Réaliser le programme Python permettant d'ouvrir le coffre (ou d'allumer la LED) en fonction de l'état des boutons poussoirs.